

## **Prova Pratica di Laboratorio di Sistemi Operativi** **15 febbraio 2012**

### **Esercizio 0 (“Se copiate, vi cacciamo”)**

Rendete la vostra directory home inaccessibile ad altri utenti (sia in lettura che in esecuzione). Rimuovete tutti i file che vi appartengono dalla directory /public.

### **Esercizio 1: Linguaggio C (obbligatorio): (20 punti)**

Guardate la directory /public/nbatch. Contiene alcuni script che stampano il proprio nome e i parametri e attendono un secondo. Lanciatene uno per prova. Scrivere un programma C che esegua tutti gli script in ordine crescente numerico. Gli script che hanno lo stesso numero devono essere eseguiti in concorrenza passando a tutti gli stessi parametri:

L'output deve essere:

```
$ ./countrun /public/nbatch -a -b -c
```

```
./00a -a -b -c
```

```
... attende un secondo
```

```
./01b -a -b -c
```

```
./01c -a -b -c
```

```
./01d -a -b -c
```

```
... attende un secondo
```

```
./02f -a -b -c
```

```
./02e -a -b -c
```

```
... attende un secondo
```

```
./10g -a -b -c
```

```
... attende un secondo
```

```
./20i -a -b -c
```

```
./20h -a -b -c
```

```
... attende un secondo
```

(l'ordine degli output dei programmi con lo stesso numero puo' cambiare).

**NOTA: e' vietato usare le funzioni system o popen.**

### **Esercizio 2: Linguaggio C opzionale: (10 punti)**

Estendere la soluzione dell'esercizio 1: l'output oltre a venir visualizzato deve essere salvato in un file nella directory corrente avente come nome il numero d'ordine degli script.

L'output del nuovo comando:

```
$ ./countrunx /public/nbatch -a -b -c
```

deve essere identico a quello di prima ma al termine devono esistere 5 file chiamati 00, 01, 02, 10, 20.

```
$ cat 00
```

```
./00a -a -b -c
```

```
$ cat 01
```

```
./01b -a -b -c
```

```
./01c -a -b -c
```

```
./01d -a -b -c
```

```
ecc.
```

### **Esercizio 3: Script bash o Python: (10 punti):**

Creare uno script o un programma python in grado di creare file con contenuti casuali e sequenze di byte nulli: l'eseguibile risultante deve avere un numero variabile di parametri, il primo e' il nome del file, il secondo e' l'ampiezza del file, i successivi indicano l'inizio e la lunghezza delle sequenze nulle. I dati sono espressi in numero di blocchi da 4K. Per esempio:

```
$nullfile nul1 100 2:1 10:4
```

deve creare un file nul1 di 400K (409600) e ha 5 blocchi interamente nulli: il terzo (dal byte 8192 al byte 12287) e quelli dall'undicesimo al quattordicesimo (dal byte 40960 al byte 57343).

Hints: per creare contenuti casuali si puo' leggere il device /dev/urandom, per inserire zeri si puo' usare il device /dev/zero. Per chi vuole scrivere uno shellscript si consiglia la lettura del manuale dd.

### **Esercizio 3 (“Consegnate! E' ora!”):**

Consegnare lo script e il sorgente del programma C, in attachment separati, entro il tempo a disposizione, via e-mail a: [renzo\\_chiocciola@cs.unibo.it](mailto:renzo_chiocciola@cs.unibo.it). Il subject del mail deve essere uguale a **PROVAPRATICA**, i nomi dei file in attachment **devono contenere il vostro cognome** (per evitare confusioni in fase di correzione).

**INOLTRE:**

Se volete che il vostro lavoro venga giudicato, lasciate aperta la vostra sessione (incluso il vostro editor) e lasciate il laboratorio. Verrete richiamati uno alla volta per una breve discussione sul vostro elaborato.